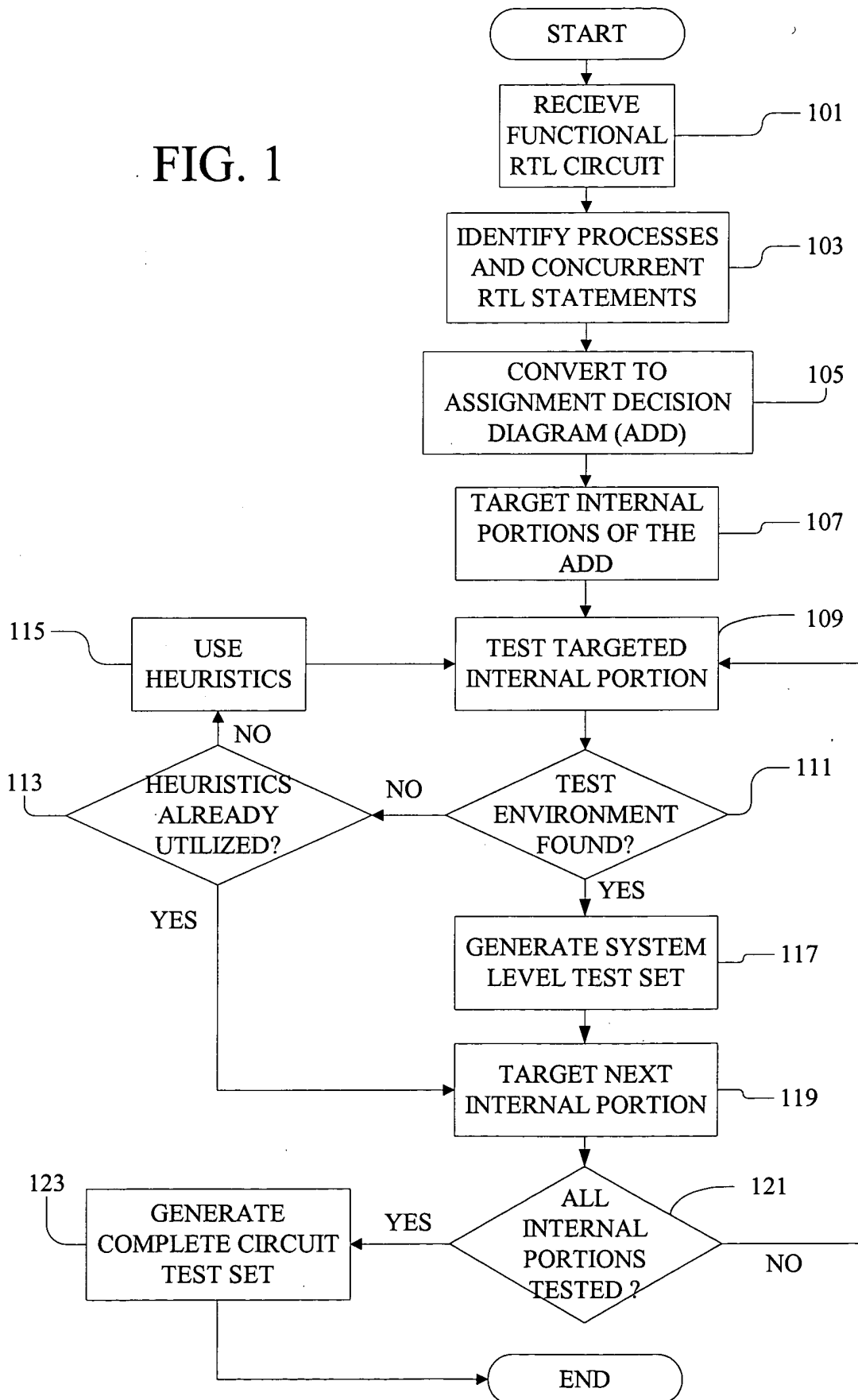


FIG. 1



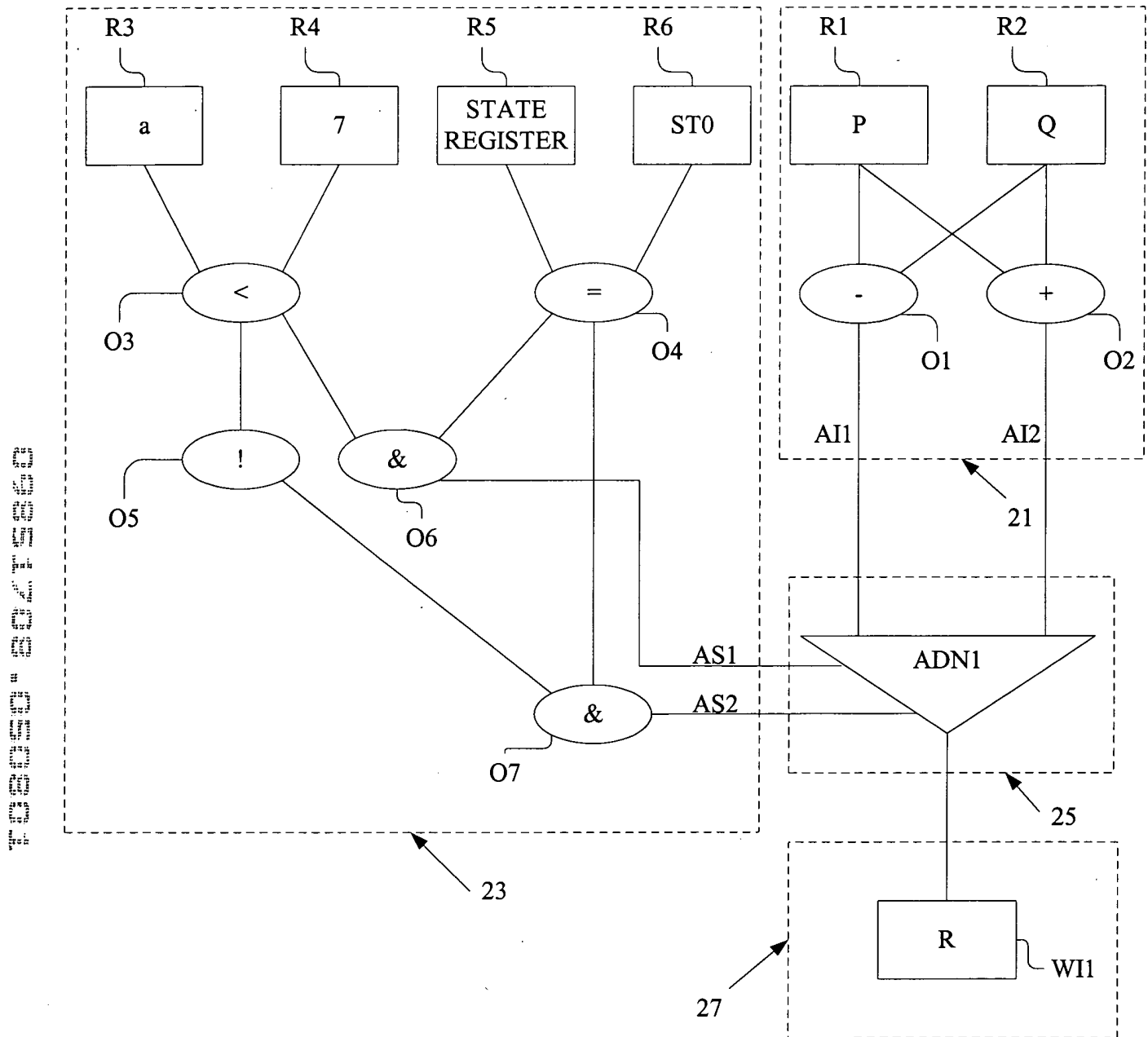


FIG. 2

```

entity T1 is
  port ( RESET, CLK :IN std_logic;
        APORT, BPORT, CPORT, DPORT : IN std_logic_vector (7 downto 0);
        E : IN std_logic;
        OPORT : OUT std_logic_vector (7 downto 0) );
  end T1;
architecture RTL of T1 is
  type STATE_TYPE is ( S0, S1, S2, S3);
  signal CURRENT_STATE, NEXT_STATE : STATE_TYPE;
  SIGNAL A, B, C, D, F, G, O, NEXT_A, NEXT_B, NEXT_C, NEXT_D,
        NEXT_F, NEXT_G, NEXT_O : std_logic_vector (7 downto 0);
  begin
    COMBIN : process (CURRENT_STATE)
      begin
        NEXT_A <= A; NEXT_B <= B; NEXT_C <= C; NEXT_D <= D;
        NEXT_F <= F; NEXT_G <= G; NEXT_O <= O; OPORT <= O;
        case CURRENT_STATE is
          when S0 =>
            NEXT_A <= APORT; NEXT_B <= BPORT;
            NEXT_C <= CPORT; NEXT_D <= D;
            NEXT_G <= "00000000";
          when S1 =>
            if ( C < D ) then
              NEXT_F <= A + B;
              NEXT_STATE <= S2;
            else
              NEXT_F <= A - B;
              NEXT_STATE <= S3;
            end if;
          when S2 =>
            NEXT_G <= F + G;
            if ( E = '0' ) then
              NEXT_STATE <= S1;
            else
              NEXT_STATE <= S3;
            end if;
          when S3 =>
            NEXT_O <= G - 3;
            NEXT_STATE <= S0;
          end case;
        end process;
    SYNCH : process (CLK, RESET)
      begin
        if ( CLK'event and CLK = '1' ) then
          if ( RESET = '1' ) then
            CURRENT_STATE <= S0;
          else
            A <= NEXT_A; B <= NEXT_B; C <= NEXT_C; D <= NEXT_D;
            G <= NEXT_G; F <= NEXT_F; O <= NEXT_O;
          end if;
        end if;
      end process;
  end RTL;

```

301

FIG. 3

CLOCK-EDGE TESTING
EXPRESSION

FIG. 4

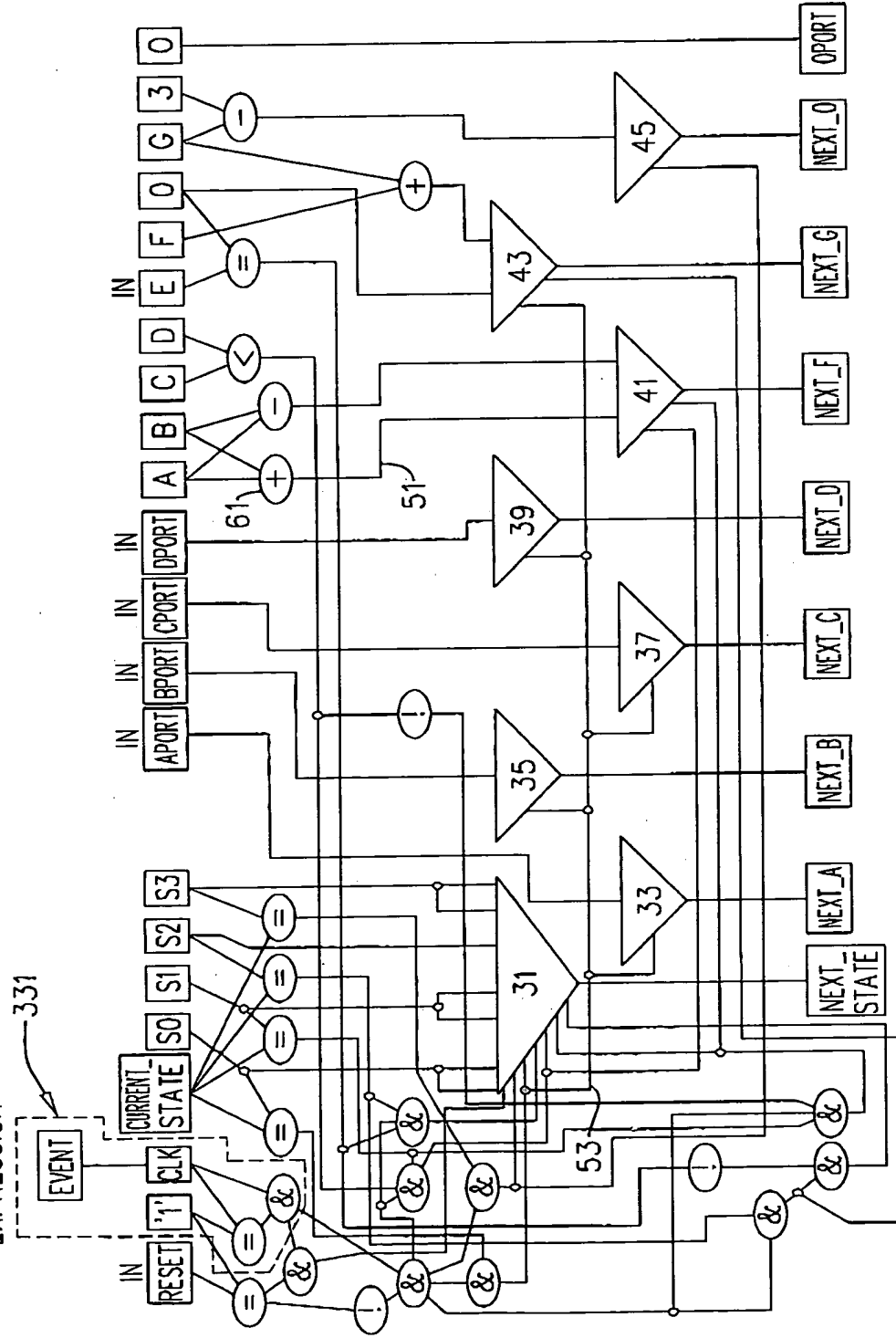


FIG. 5

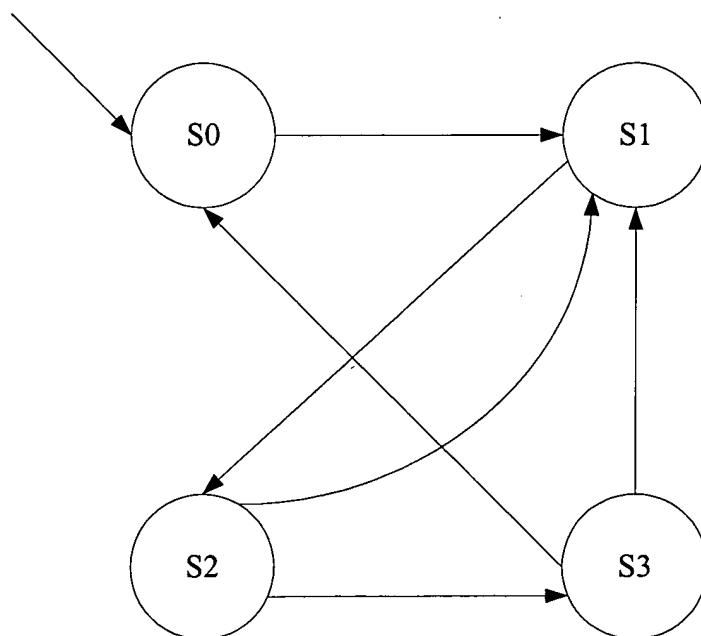


FIG. 5

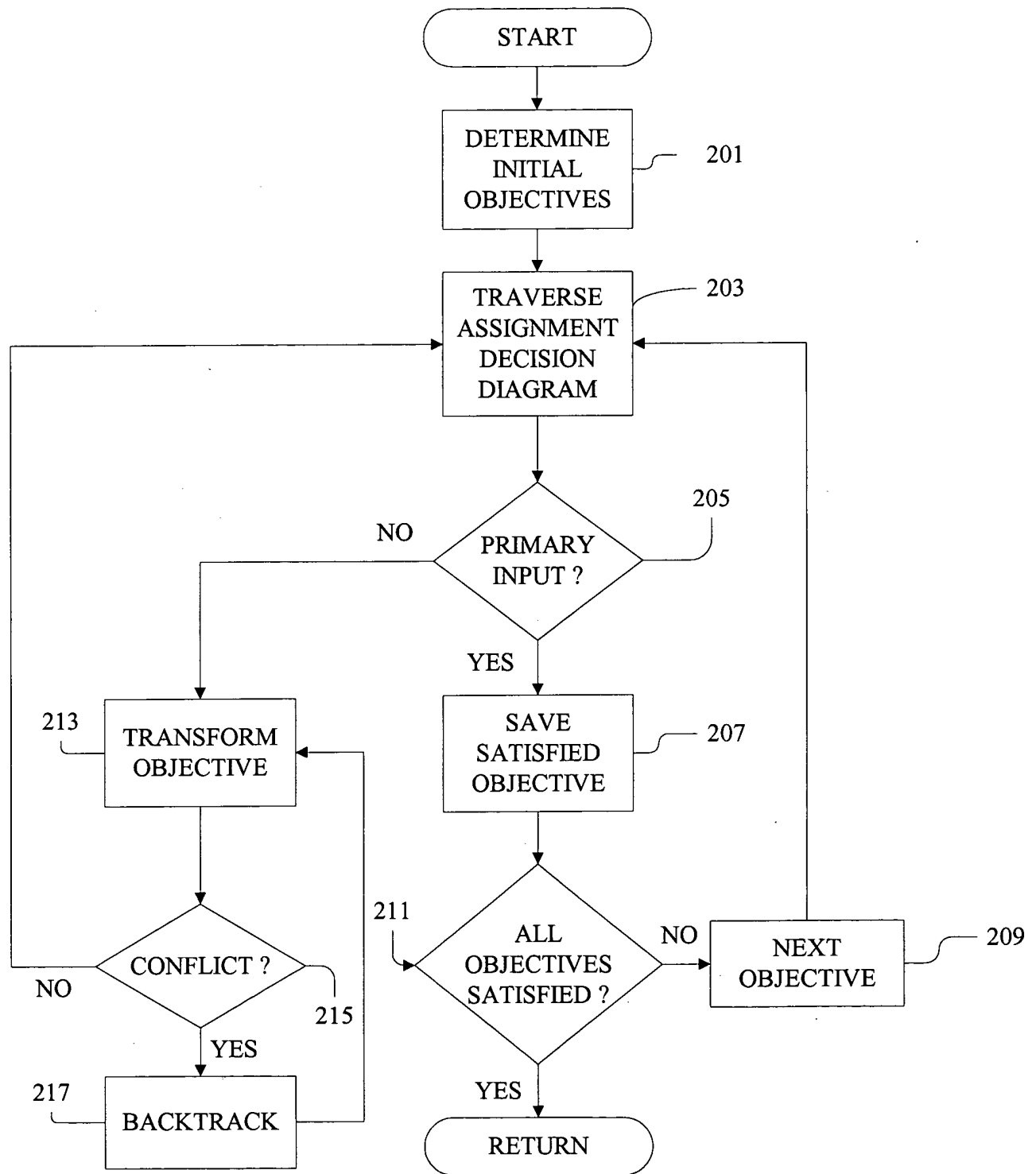


FIG. 6

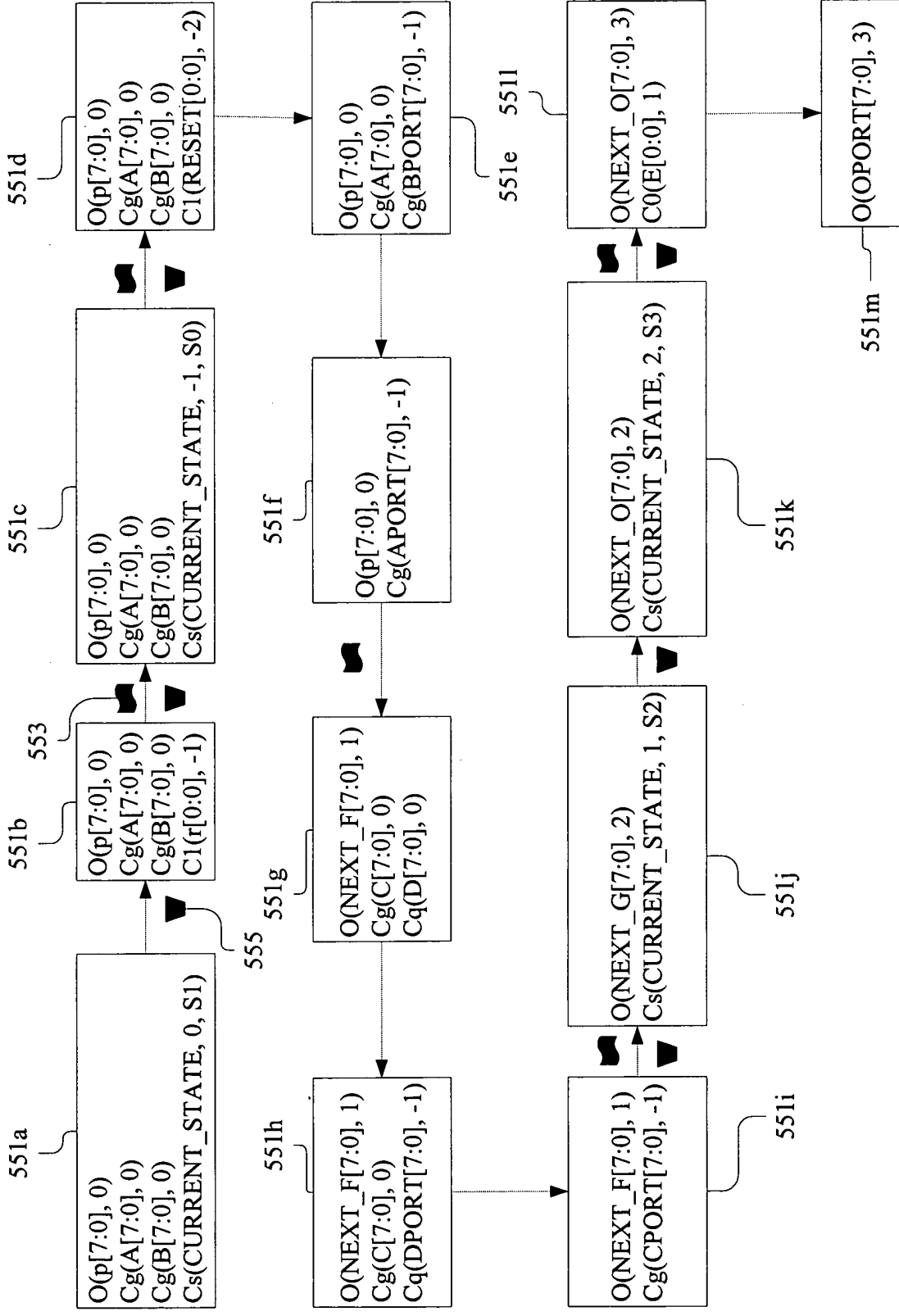


FIG. 7

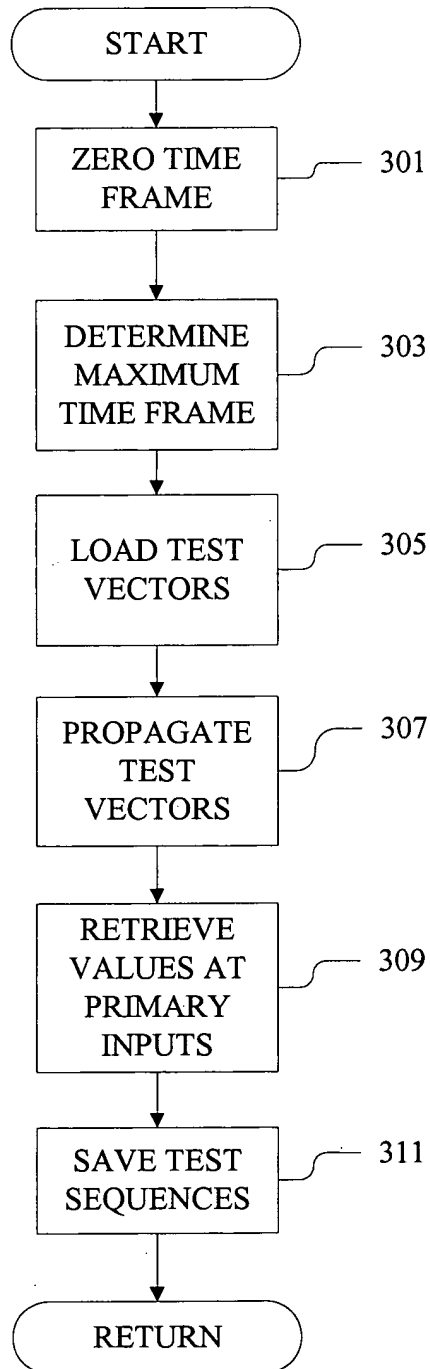


FIG. 8

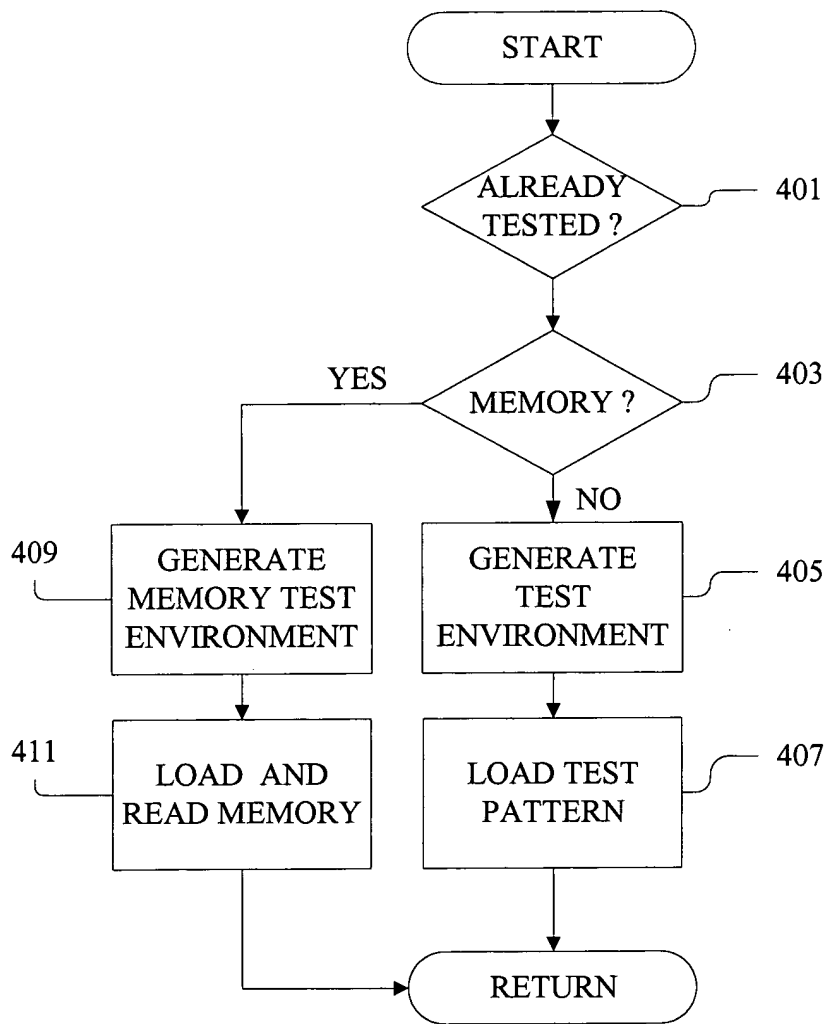


FIG. 9

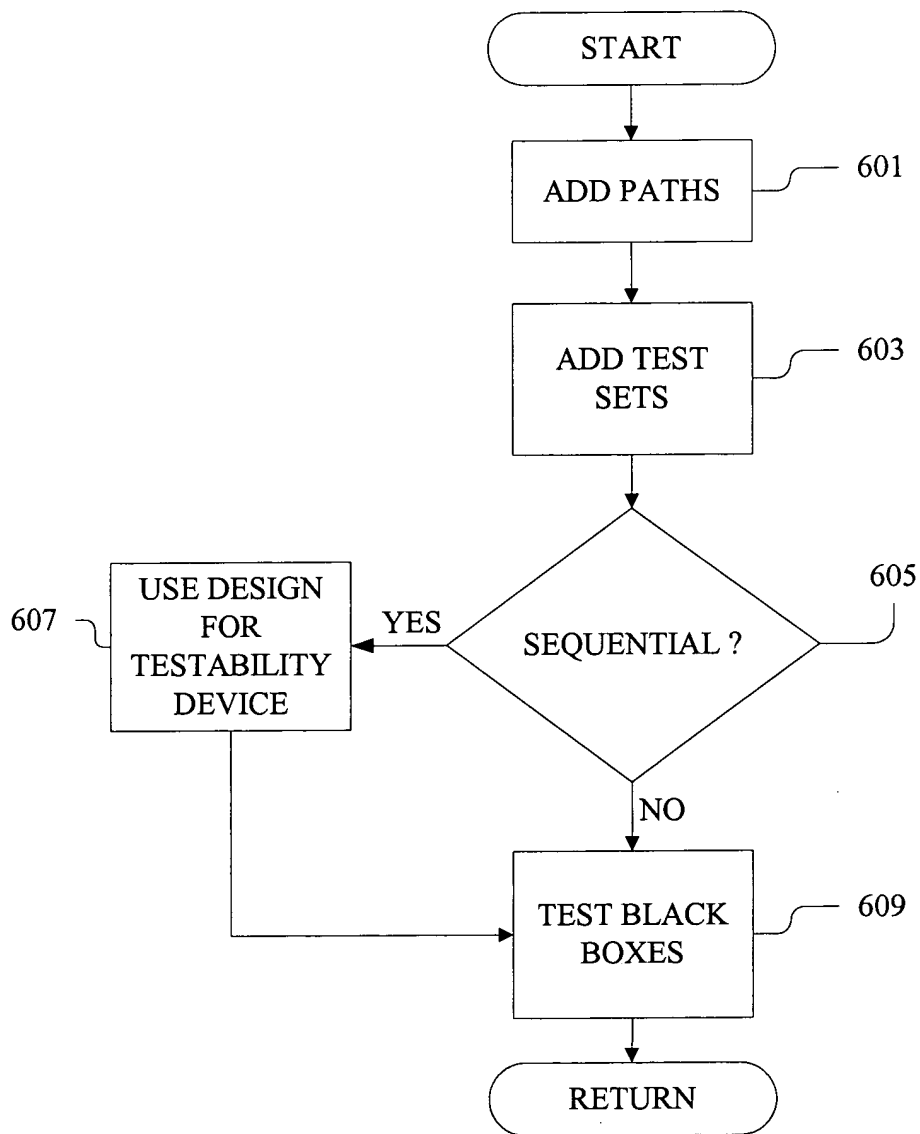


FIG. 10